TFAWS Paper Session

Potential of the Spectral Element Method in Flow Simulations of Aerospace Systems

K.E. Davis and B.C. Houchens

Presented By
Kenneth Davis

Thermal & Fluids Analysis Workshop
TFAWS 2010
August 16-20, 2010
Houston, TX

Overview

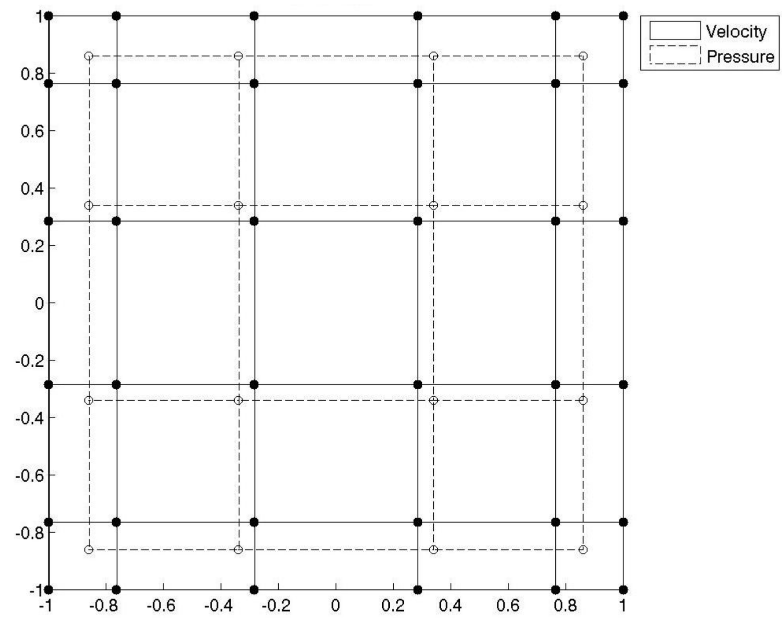
- Code development
  - Utilizes the spectral element method to solve incompressible fluid flow and heat transfer equations
  - Written from scratch
  - Can handle complex geometries
  - Arbitrary application of boundary conditions
  - Several typical boundary conditions
- Advantages over commercial software
  - Total control
  - Application of “unusual” boundary conditions
  - More accuracy
  - More cost-effective

Why Use Spectral Elements?

- Accuracy
  - Can refine in p as well as h to improve accuracy
  - Finite elements and finite volumes are usually limited to h refinement
  - p refinement yields better results than h refinement
- No need for stabilization
  - Finite elements generally use elements (such as linear-linear) that require stabilization
  - Spectral elements are stable when using the $P_N – P_{N-2}$ grids
- Can handle complex geometries
  - Finite difference methods are limited to simple domains

The Spectral Element Method

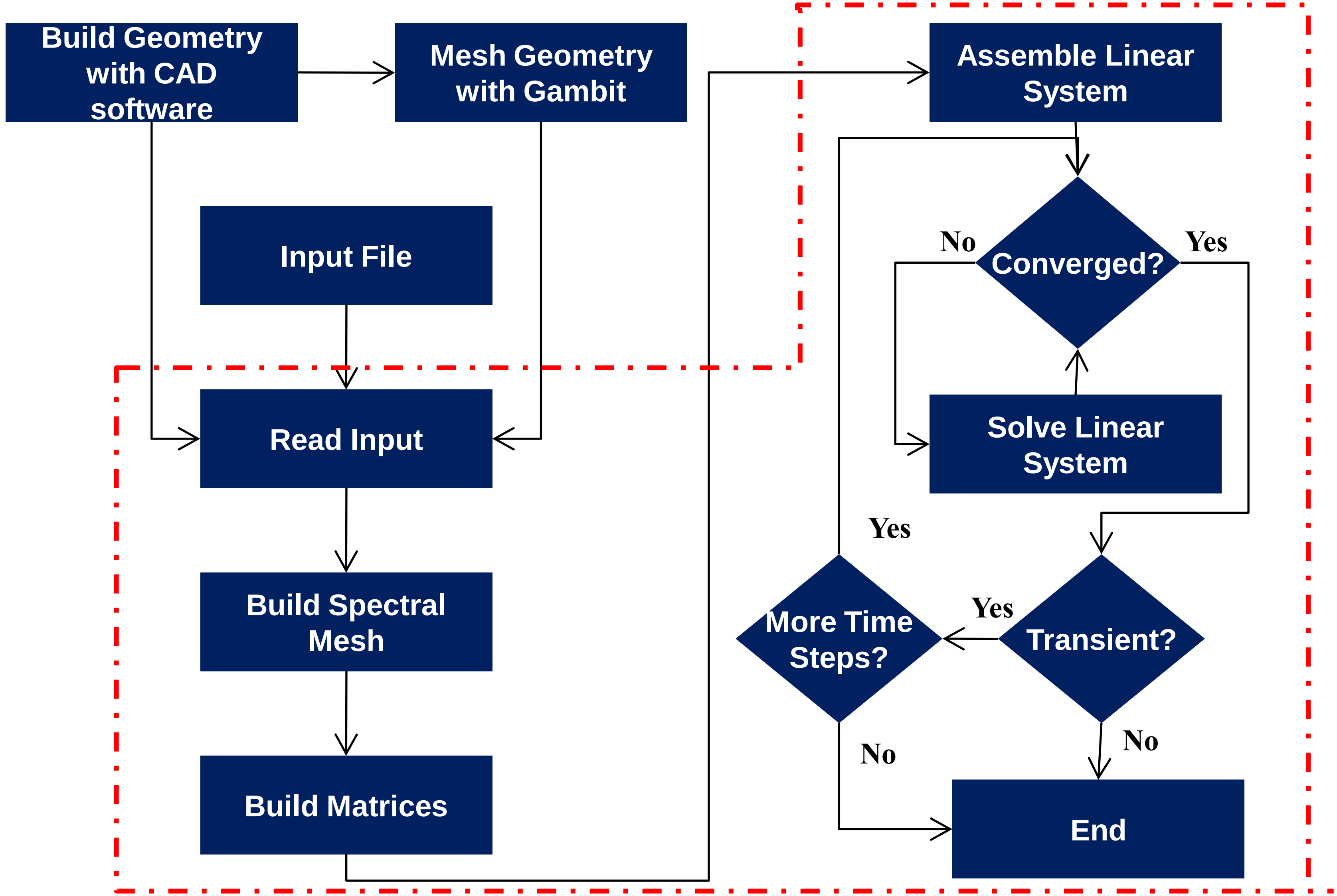
- Subset of the finite element method
  - Difference is in the definition of the basis functions
- $P_N - P_{N-2}$ Grid
  - Velocity is solved on a Gauss-Legendre-Lobatto grid of order N and pressure is solved on a Gauss-Legendre grid of order N-2
  - Satisfies the Babuska-Brezzi condition
  - Basis functions are Lagrange interpolants through all nodes on the grid
- Galerkin approximation is used for weighting functions



The Spectral Element Method

- Discretize the domain
  - First using meshing software such as Gambit
  - Then build the spectral mesh on each element
- Approximate solution
  - $$u^e = \sum_{i}^{nn_e} \psi_i(x,y) u_i$$
- Procedure
  - Multiply by test function
  - Integrate over each element
  - Scatter to global matrices
  - Newton-Raphson iterations
    - Solve the resulting linear system using GMRES or BiCGStab
  - Write data

Solver Structure



Code Input and Output

Input File

```
----------RUN PARAMETERS---------------
problem_type        fhtc
coordinate_frame    cartesian
transient|steady    transient
Nu                  12
Np                  10
mesh_file           cavity.neu
geometry_file       cavity.igs

----------SOLVER PARAMETERS------------
NR_iterations       10
linear_solver       bicgstab
linear_iterations   2000
restart_gmres       2000
MGscheme            86468
left_precond        ilu0
right_precond       ilu0
tolerance           1e-12
initialization      value
UVPT_init_values    0.0 0.0 0.0 0.0
---------transient parameters----------
numberoftimesteps   100
time_step_size      0.1
order               3
num_ts_bef_output   1

----------FLUID PROPERTIES-------------
viscosity           1.0
density             100.0
conductivity        1.0
specific_heat       1.0
--------buoyancy---------
buoyancy&direction  off  x
boussinesq          off
ref_density         1.0
ref_temperature     0.0
thermal_exp_coeff   .05
----------mhd------------
mhdforce            off
Hartmann_Number     50.0

----------SCALING PARAMETERS-----------
Scaling_type        viscous
Length_Scale        1.0
Velocity_Scale      1.0
Temp_Diff_Scale     1.0
ref_temperature     0.0

---------BOUNDARY CONDITIONS-----------
number_bnds         2
------------Flow Conditions------------
boundary1           velocity  0.0  0.0
boundary2           velvarfc  if (t<1) -t else -1.0  0.0
------------Energy Conditions----------
boundary1           temperature  0.0
boundary2           temperature  0.0
```

Output File

```
Fluid flow coupled with thermal energy.

Simulation parameters
---------------------------------------
Density:                    100.0000
Viscosity:                1.0000E+00
Thermal Conductivity:         1.0000
Specific Heat:                1.0000
---------------------------------------
Transient parameters
---------------------------------------
Method:      Backwards Differentiation
Order:                               3
Number of Time Steps:              100
Time Step Size:                 0.1000
---------------------------------------

System Data
-----------------------------------------------
Number of X Momentum Equations :          264
Number of Y Momentum Equations :          264
Number of Continuity Equations :          242
Number of Energy Equations     :          264
Number of Total Equations      :         1034
-----------------------------------------------


Time Step    1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

----------------------------------------------------------------
| Init. Residuals |       RMS        |        Max        |
----------------------------------------------------------------
| X Momentum      |    2.20965E+01   |    7.20983E+01    |
| Y Momentum      |    7.87765E-01   |    6.60260E+00    |
| Continuity      |    1.76589E-02   |    1.21737E-01    |
| Energy          |    0.00000E+00   |    0.00000E+00    |
----------------------------------------------------------------
| System          |    1.11723E+01   |    7.20983E+01    |
----------------------------------------------------------------

Nonlinear Iteration            1
****************************************************************

----------------------------------------------------------------
| Residuals       |       RMS        |        Max        |
----------------------------------------------------------------
| X Momentum      |    8.63221E-02   |    6.55951E-01    |
| Y Momentum      |    6.36657E-02   |    2.36447E-01    |
| Continuity      |    6.80520E-03   |    3.05551E-02    |
| Energy          |    0.00000E+00   |    0.00000E+00    |
----------------------------------------------------------------
| System          |    5.42977E-02   |    6.55951E-01    |
----------------------------------------------------------------
```

Current Code Capabilities

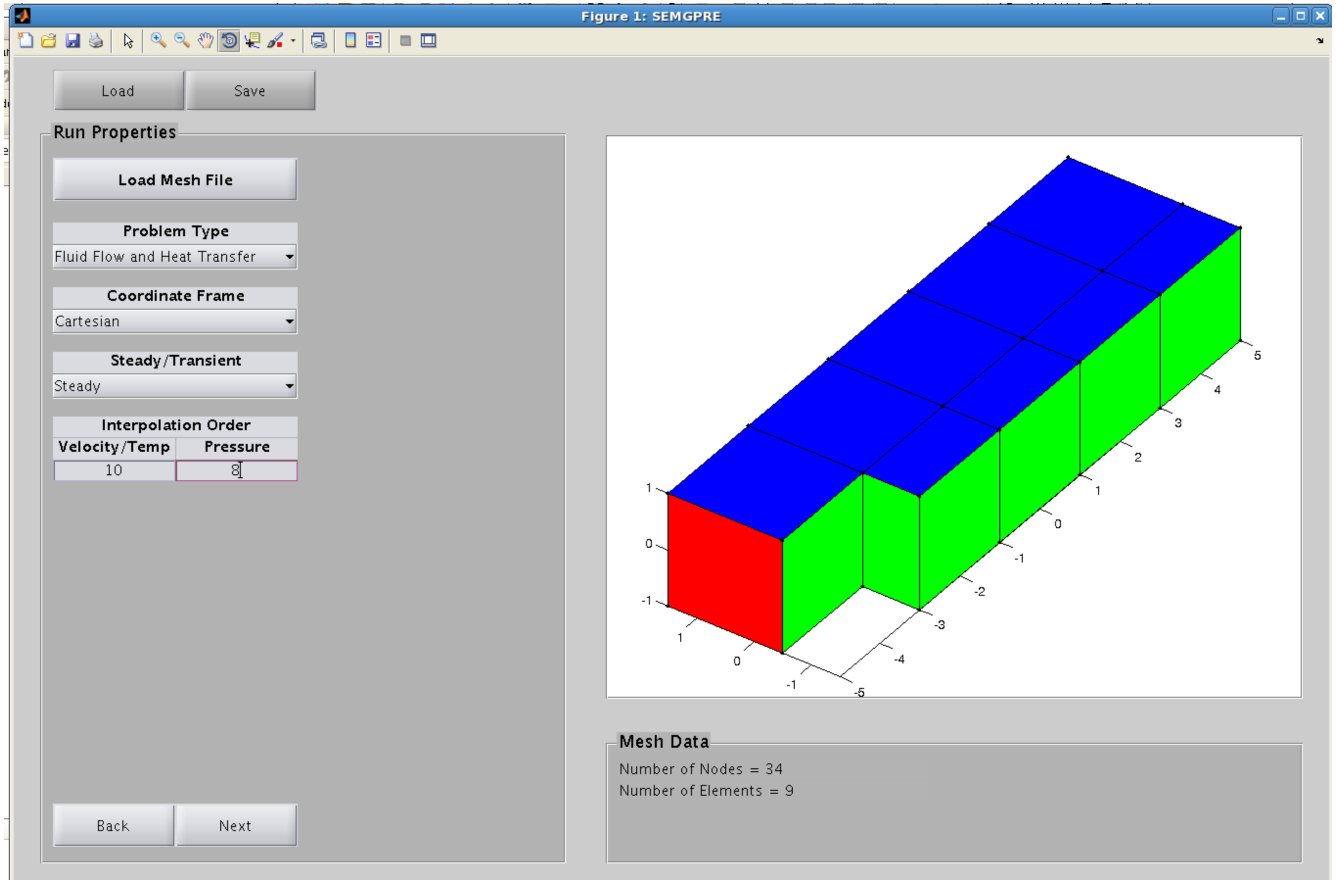
- General geometries are represented exactly (2D only)
  - Code reads IGES files and stores geometry parameters for each curve
  - Must find where mesh and geometry coincide
  - Allows for exact computation of Jacobian
- Boundary conditions can be applied to any boundary
  - Fluid boundary conditions
    - Velocity components
    - Stress components
    - Mixed velocity/stress components
  - Thermal boundary conditions
    - Temperature
    - Heat flux
  - All boundary conditions can vary with space
  - Velocity and temperature can vary with time

Current Code Capabilities

- Initial conditions can vary with space
- Cartesian – 2D and 3D, Cylindrical – 2D only
  - 2D cylindrical coordinates refers to axisymmetric flows, meaning the coordinates are r and z
  - Currently extending the 3D code to solve in cylindrical coordinates
- High-order transient solutions
  - Attempted Adams-Moulton method, but it was unstable
  - Now use backwards differentiation up to 6th order
- Buoyancy
  - Boussinesq approximation can be applied
    - $\rho = \rho_{ref}[1 - \alpha(T - T_{ref})]$

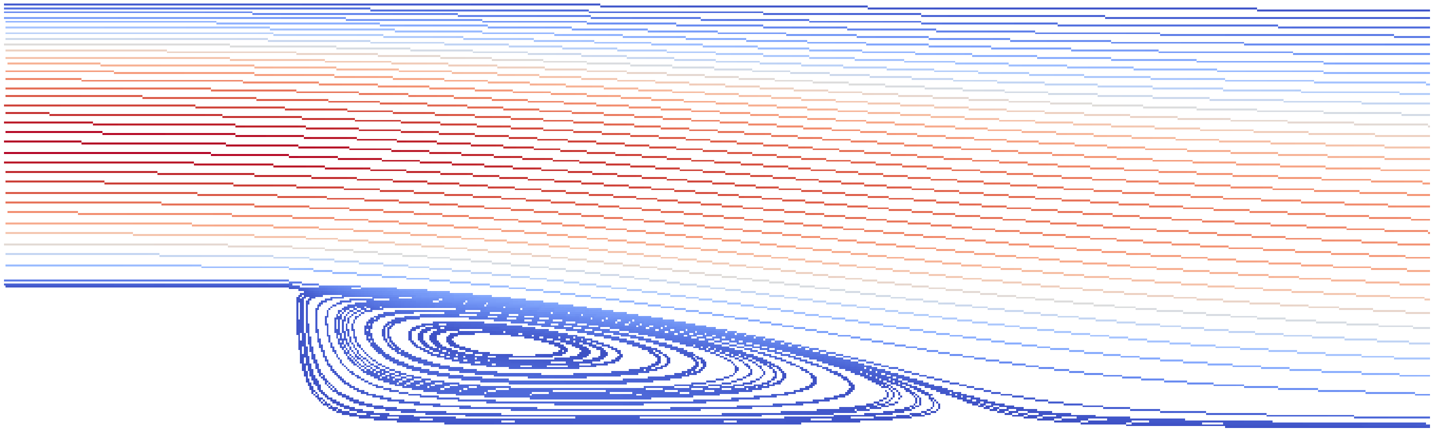
Pre-processing Matlab GUI

- Pre-processor writes input file for code
- Provides a simple interface for users unfamiliar with the code and its input file



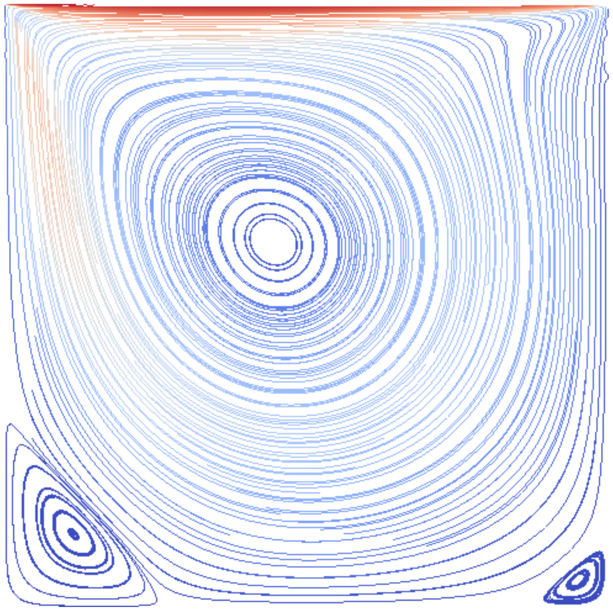
Backward-Facing Step

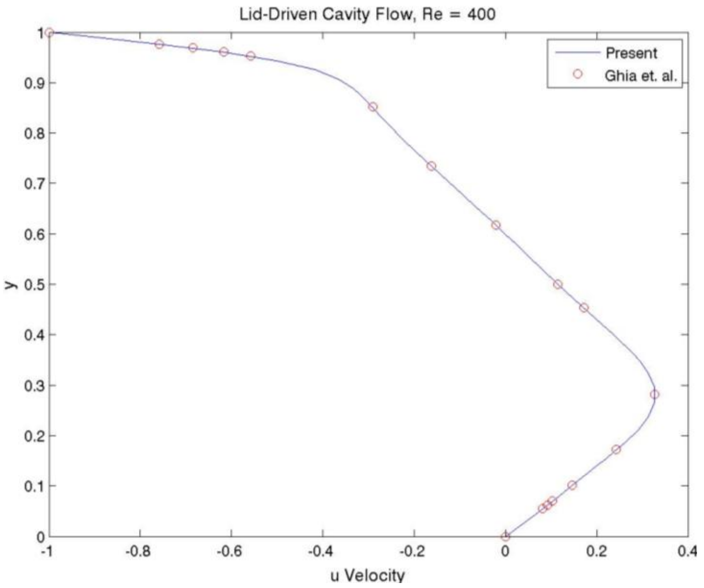
- Common benchmark problem
- Re = 109.5 – used by A.T. Patera in his 1984 paper introducing spectral elements
- Reattachment occurs at $L_r \approx 5.0$ as expected
- Recirculation at the channel expansion is seen



Lid Driven Cavity Flow

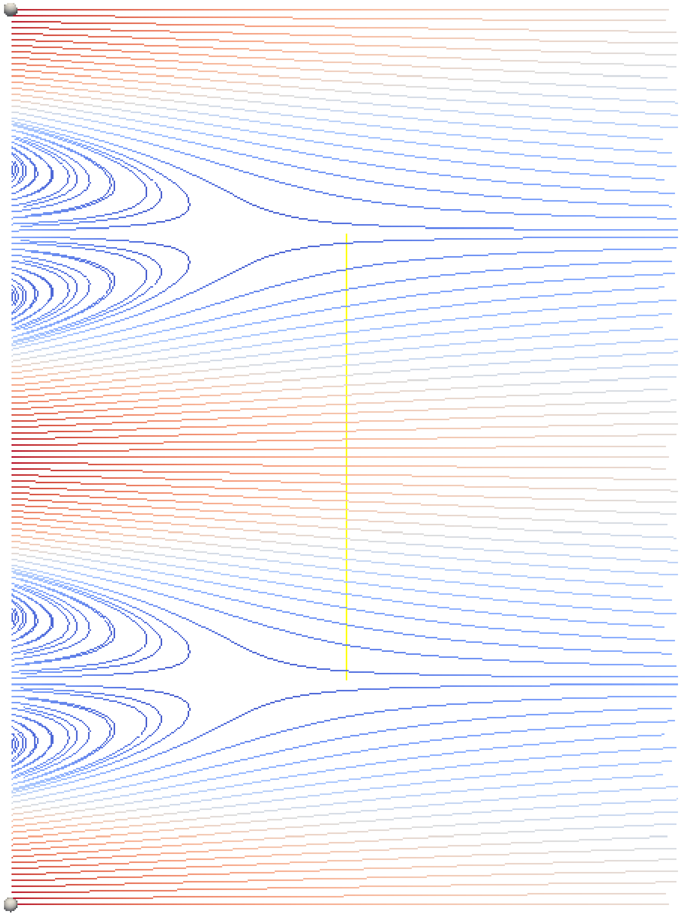
- Re = 400
- Top side has dimensionless velocity of 1 to left; all other sides are at rest
- Recirculations qualitatively accurate and the u velocity on the vertical centerline agrees well with previous results





Kovasznay Flow

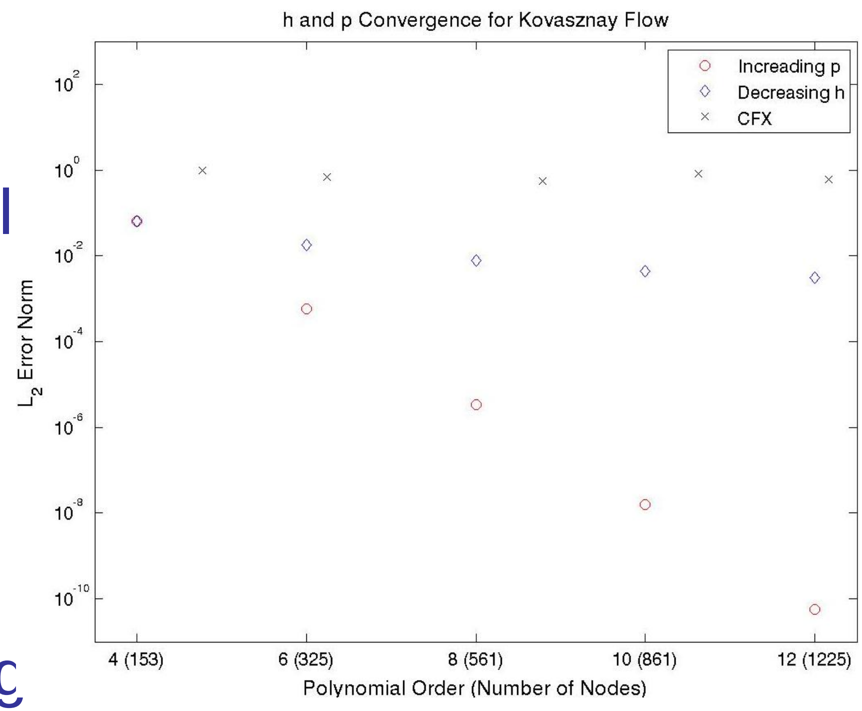
- Flow behind a two dimensional grid
- Exact solution given by L.I.G. Kovasznay in 1948
  - $u(x, y) = 1 - e^{\lambda x} cos(2\pi y)$
  - $v(x, y) = (\lambda/2\pi) e^{\lambda x} sin(2\pi y)$
  - $p(x) = (1 - e^{2\lambda x})/2$
  - $\lambda = Re/2 - (Re^2/4 + 4\pi^2)^{1/2}$
- *Re = 40* for this simulation
- Dirichlet boundary conditions were applied



- Obtained a solution where the L_2 norm of the error in velocity is less than 10^{-10}

Convergence Results

- Simulations were run until the root-mean-square of the fully coupled system residual reached 10^{-13}
- Increasing the polynomial order gives much more accurate results than decreasing the mesh spacing



- At P = 12, the P simulation is 8 times more accurate
- Comparison to the commercial code Ansys CFX 12.0
  - Uses a finite volume method and linear-linear elements
  - The accuracy of CFX is much less than our own code
  - In both codes, h refinement shows little impact on accuracy

Current Activity

- Linear solver
  - Currently solve fully coupled system using ILU(0) preconditioned Krylov subspace methods
  - Implemented multigrid, but not to satisfaction
- Preconditioning
  - Currently use ILU(0), but may need something more parallelizable in the future

Linear solver and preconditioning tests

Time (s)	None	Diagonal	ILU(0)
GMRES	160.7	70.8	7.1
BiCGStab	141.4	33.6	7.2

Iterations	None	Diagonal	ILU(0)
GMRES	1728	1109	63
BiCGStab	3498	793	45

Current Activity

- Adding additional boundary conditions
  - Normal/tangential conditions, convection (by coefficient), rotational velocity, translational and rotational periodic conditions
- Adding spatial variability to fluid properties
- Parallelizing subroutines
- Extending to 3D
  - Adding all features that are included in 2D code
  - Allowing for cylindrical coordinates
  - Handling all geometries
- Writing a post-processor
  - Currently use Gambit for all post-processing
  - Will compute derivatives of all variables, streamfunction, vorticity, and will integrate any variable over any surface

Future Goals

- Parallelization of entire code
- Represent all 3D geometries exactly
- Turbulence modeling
  - Basic two-equation models
    - $k - \epsilon$
    - $k - \omega$
  - Large eddy simulation
- Solve compressible flow equations

Conclusion

- The spectral element method is an effective method for solving fluid flow and heat transfer problems
- Our in-house code has been benchmarked for several 2D cases, but still needs 3D benchmarking
- p refinement yields more accurate results than h refinement
  - This accuracy makes the spectral element method more attractive than basic finite elements
- Commercial codes like Ansys CFX do not use the spectral element method and, consequently, are limited in accuracy

References

[1] Patera, A. T. (1984) *J. Comp. Phys., 54,* 468–488.

[2] Ghia, U. et al. (1982) *J. Comp. Phys. 48*, 387-411

[3] Kovasznay, L. I. G. (1948) *Proc. Cambridge Philos. Soc., 44,* 58–62.